ATIGS
0.1V

ATIGS(Automated Test Input Generation Software)

The purpose of this document is to provide with a template for documenting ATIGS

Document Control :

Project Revision History

Date	Version	Author	Brief Description of Changes	Approver Signature
23/11/22	0.1	Group 01		

1. Introduction

Automated Test Input Generator Software (ATIGS) is developed to automatically generate test data for application.Manually creating test data for each and every application can be labor-intensive. This ATIGS helps in creating test data for multiple applications.

1.1. Intended Audience

This document could be shared or viewed across all the following members CG employees, BU SME's. internal SME's.

This is a technical document, and the terms should be understood by all of them.

CG Employee	
BU SME'S	
Internal SME's	

1.2.Acronyms/Abbreviations

ATIGS	Automated Test Input Generation Software

1.3.Project Purpose

The Automated Test Input Generator Software [ATIGS] generate a number of tests to check whether programs meet requirements or not.

1.4.Key Project Objectives

To automatically create test data files for different applications.

1.5.Project Scope and Limitation

1.5.1 In Scope

The purpose of this application is to automatically generate test data for an application.

1.5.2 Out of scope

It would be difficult to handle a large number of files and to provide a unique name to each one of them.

1.6.Functional Overview

The ATIGS basically accepts an input file with specified range , and then automatically generates different test cases in output files.The software first prompts to enter the number of output files required and the input file itself.

It then reads the file for parameter name , data type and value and tests it against the given range.

It also identifies comments in the input file and detects errors in the input file format.

Then generates the output file with given name-value pairs.

1.7 Benefits of ATIGS

1.In case of different functions involving mathematical operations, user may need to input same data or combinations of data multiple times. In such scenarios , these automatically generated test cases play an important role.
2.The different functionalities and application authentication can be easily tested using these test case files.
3. It helps the developer to check their applications from all aspects to detect failures.

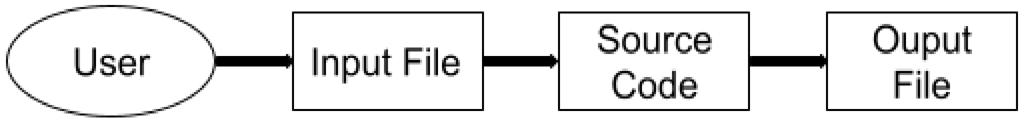
2 Design Overview

2.1 Design Objectives

The goal of ATIGS is to automatically generate test cases for an application.

2.1.1 Recommended Architecture

The recommended system architecture is as follows.



- Initially, user will give input in command line
- Then system will read that input file
- It will check input file for validation
- If system validates input file after checking, it will generate test data accordingly
- The generated data will be shown in output file created

2.2 Architectural Strategies

2.2.1 Reuse of Existing Common Services/Utilities

The project does not reuse any new common services or utilities.

2.2.2 Creation of New Common Services/Utilities

The project does not create or use any new common services or utilities.

2.2.3 User Interface Paradigms

- Desktop or a Linux machine with internet connection.
- Command Line Interface (CLI).

2.2.4 System Interface Paradigms

- Operating system – Unix.
- Linux Kernel version - 4.4.0-19041-Microsoft.
- Bash shell: x86_64 GNU/Linux

- **2.2.5 Function Declarations**
  - void ReadInputFile()
    To read input file and display error for wrong input.
  - void ReadSettings()
    Opens Settings.config file in read mode and displays error when not present.
  - int CheckFileForValidity()
    Checks for parameter name and data type and any comment, if present.
  - void Printfile()
    Creates Output and and writes in it.
  - void CompareBool()
  - void CompareInt()
  - void ComapreFloat()
  - void CompareString()
  - void DispError(int ErrorCode)
    To display error in command line input and input file format.
  - void CreateFile()
    Creates output file and open in read/write mode.
  - void WriteFile()
    Writes required data in the output files.

2.2.6 Error Detection / Exception Handling

- Appropriate error message with line number for file handling will be included

2.2.7 Memory Management

NA

2.2.8 Performance

Quick response

System will detect errors at every step instead of showing errors at the last step.

2.2.9 Security

No user is allowed to make changes in the output files or disturb the contents/test cases.

2.2.10 Concurrency and Synchronization

NA

2.2.11 Housekeeping and Maintenance

- After the creation of output files once , the program overwrites the same files when re-run.
- So , it does not keep filling up the memory instead re-uses them each time.

3.Experience System Architecture

ATIGS is software for a single user experience and does not use any server , just the disk space of the system.

3.1 System Architecture Diagram. (Not Necessary)

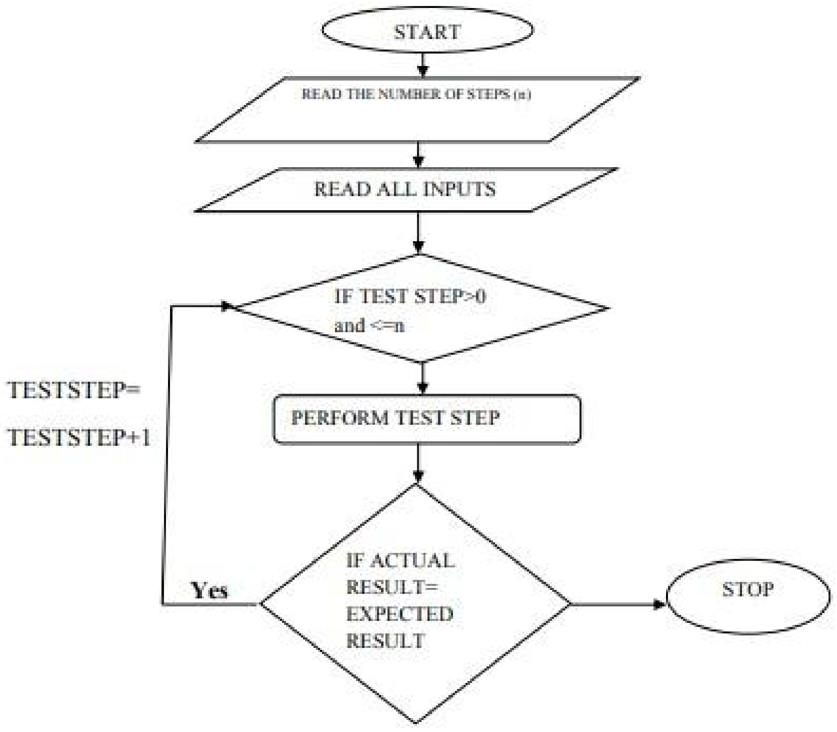
NA

3.2 System Use-Cases

- User runs the program in the linux interface.
- The program runs and prompts user to enter the desired inputs along with input file.
- After the program successfully runs and produces desired output , the user can access the output files from Output_directory.

3.3 Subsystem

- Design test cases- Purpose is to generate and prepare a set of test cases.
  Outcome is set of test cases.
- Prepare test data- Purpose is to generate and prepare test data for each test case.
  Outcome is a set of test data.
- Run program with test data- Test cases and test data will run in this step.
  Result is actual system output.
- Compare results to test cases- This step is used to compare the system output to expected output in the test case. Result is test report of running the test cases with test data.



Test case generation process

3.4 System Interfaces

3.4.1 Internal Interfaces

The internal interfaces comprise interfaces through which the system interacts with the user through which it provides them services.

- Linux Interface

  The user is required to login into a linux operating system to access the program and the desired output.

- Internet : It is not mandatory to have a connection for running this software.

3.4.2 External Interfacess

The external interface comprises interfaces through which the users interact with the system.

- Desktop or Linux Machine
- ATIGS software in the system.

4. Detailed System Design

ATIGS will take data from input file and will produce data in output file according to name-value pair.

4.1 Key Entities

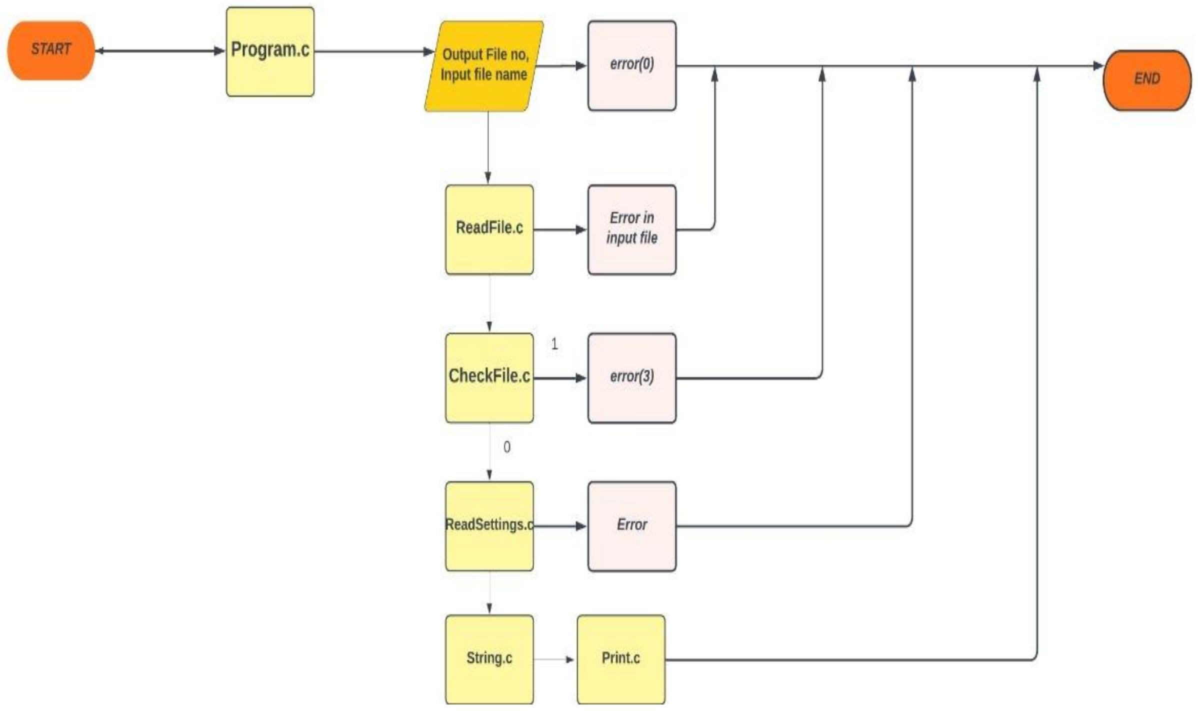
The key entities associated with the system are:-

User

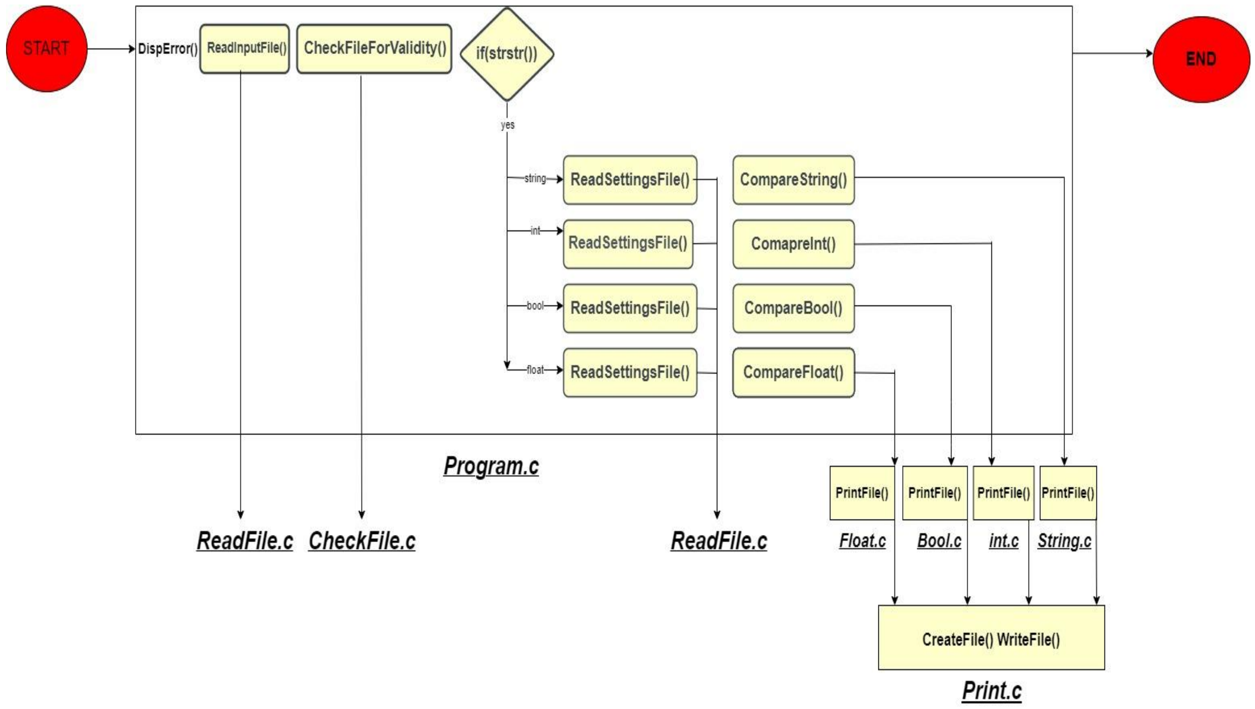
- User is an entity which ask for test data generation.
- User gives number of output files to be generated for testing data

4.2 Detailed-Level DataFlow Diagram

4.2.1 Level - 1 Diagram



4.2.2 Level -2 Diagram



4.3 Archival and retention requirements

The system shall allow user to give input file name and produce output file accordingly.

4.4 Disaster and Failure Recovery

NA

4.5 Business Process workflow

Later

4.6 Business Process Modeling and Management (as applicable)

Later

4.7 Business Logic

Later

4.8 Variables

NA

4.9Activity / Class Diagrams (as applicable)

Later

4.10 Data Migration

Data is migrated from input file to output file.

4.10.1 Architectural Representation

Later

4.10.2 Architectural Goals and Constraints

Later

4.10.3 Logical View

Later

4.10.4 Architecturally Significant Design Packages

Later

4.10.5 Data model

Later

4.10.6 Deployment View

Later

5.Environment Description

The environment description allows the user to take a number of output files and a test input file as arguments.

5.1 Language Support

C language and compilation using gcc. Shell script to execute the program.

5.2 User Desktop Requirements

User desktop requires a Linux environment, Operating system of Linux Debian or Ubuntu 20.04.5 LTS (GNU/Linux 4.4.0-19041-Microsoft x86_64) kernel version and reliable internet connectivity.

5.3 Configuration

NA

5.3.1 Operating System

- Operating system – Linux
- RAM - 8GB.

- Processor - i3/i5/i7.

5.3.2 Database

NA

5.3.3 Network

NA

5.3.4 Desktop

Unix like environment is required

6. References

- https://www.tutorialspoint.com/c_standard_library/c_function_atoi.htm
- https://www.tutorialspoint.com/cprogramming/c_command_line_arguments.htm
- https://devenum.com/how-to-read-text-file-word-by-word-in-c/
- https://www.geeksforgeeks.org/basics-file-handling-c/

7.Appendix

NA

QMS Template Version Control (Maintained by QA)

Date	Version	Author	Description
23/11/2022	0.1	Group 01	Initial Draft